Structured Support Exploration For Multilayer Sparse Matrix Factorization

Quoc-Tung Le, Rémi Gribonval

STRUCTURED SUPPORT EXPLORATION FOR MULTILAYER SPARSE MATRIX FACTORIZATION

Le Quoc Tung and Gribonval Rémi

Univ Lyon, ENS de Lyon, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France.

ABSTRACT

Matrix factorization with sparsity constraints plays an important role in many machine learning and signal processing problems such as dictionary learning, data visualization, dimension reduction. Among the most popular tools for sparse matrix factorization are proximal algorithms, a family of algorithms based on proximal operators. In this paper, we address two problems with the application of proximal algorithms to sparse matrix factorization. On the one hand, we analyze a weakness of proximal algorithms in sparse matrix factorization: the premature convergence of the support. A remedy is also proposed to address this problem. On the other hand, we describe a new tractable proximal operator called Generalized Hungarian Method, associated to so-called k-regular matrices, which are useful for the factorization of a class of matrices associated to fast linear transforms. We further illustrate the effectiveness of our proposals by numerical experiments on the Hadamard Transform and magnetoencephalography matrix factorization.

***Index Terms*—** Matrix factorization, Sparsity, Support Exploration, Hungarian Method, Proximal Algorithm.

1. INTRODUCTION

Sparsity is an important concept in machine learning. Indeed, sparse objects (sparse graphs, sparse matrices) naturally arise in many applicative contexts and offer an easier manipulation and exploitation in comparison to their dense counterparts computational wise. Techniques involving sparsity in inverse problems such as: Best Subset Selection ([1], [2]), Ridge Regression ([3]), LASSO ([4], [5]), Matching Pursuit ([6], [7], [8],[9]) received great attention from researchers.

In this paper, we highlight structured sparsity in the problem of multilayer matrix factorization. In fact, many important linear operators such as the Discrete Fourier Transform, the Hadamard Transform can be factorized into multiple sparse factors with specific sparse patterns. Those factorization allows fast transform algorithms (such as the Fast Fourier Transform [10]) with complexity $O(n \log n)$ instead of $O(n^2)$. Such operators can serve as a testbed to explore the capacity of sparse matrix factorization algorithms.

One possible formulation [11] of the problem of multilayer structured sparse matrix factorization is: given a matrix $A \in \mathbb{R}^{m \times n}$, find $S_j \in \mathbb{R}^{a_j \times a_{j+1}}, j = 1, \ldots, J$ with $a_1 = m, a_{J+1} = n$ such that $A \approx \prod_{i=1}^{J} S_j$. This is expressed as:

$$\underset{S_1,\ldots,S_j}{\text{Minimize}} \|A - \prod_{j=1}^{J} S_j\|_F^2 + \sum_{j=1}^{J} \delta_{\mathcal{E}_j}(S_j) \tag{1}$$

where $\delta_{\mathcal{E}_j}$ are indicator functions of sets $\mathcal{E}_j$ of structured sparse matrices. Typically, $\mathcal{E}_j$ can be the set of matrices with at most k-nonzero entries, or more generally a set of matrices whose supports (indices nonzero entries of the considered matrix) satisfies a prescribed contraint.

In [11], a proximal gradient algorithm [12] (named Proximal Alternating Linearized Minimization – PALM [13]) is employed for problem (1). It is an iterative algorithm with the following update rule:

$$S_j^{i+1} \leftarrow P_{\mathcal{E}_j}\left(S_j^i - \mu_i^j \nabla_{S_j} \|A - (\prod_{l=1}^{j-1} S_l^{i+1})(\prod_{l=j}^{J} S_l^i)\|_F^2\right) \tag{2}$$

where S_j^i is the j^{th} factor at the i^{th} iteration, $P_{\mathcal{E}_j}$ is the proximal (or here equivalently, the projection) operator onto the set $\mathcal{E}_j$, and the step size μ_i^j is chosen according to certain Lipschitz properties [11] to ensure convergence to a stationary point [13]. The update rule (2) simply consists of a gradient step followed by a proximal (projection) operator. However, experimental results showed that the performance of PALM degrades when the number of factors increases [11]. To address this problem, the author proposed a hierarchical factorization (HF), in which the matrix is only factorized into two factors. Those factors are further factorized to achieve multilayer factorization.

In this paper, we firstly analyze two difficulties with both approaches in [11]: the premature convergence of the support of the factors with PALM, which leads to its poor performance; and the possible non-factorizability of intermediate matrices with HF. Both serve as the motivations for our contributions: a remedy to help generic proximal algorithms to actively explore the support in Section 2 and a newly defined structured set $\mathcal{E}_j$ of k-regular sparse matrices with a tractable proximal operator enabling PALM to successfully factorize the Hadamard Transform –without having to resort to HF– in Section 3. Illustration for the effectiveness of our proposal will be shown by numerical experiments in Section 4.

2. BILINEAR HARD THRESHOLDING PURSUIT

When $\mathcal{E}_j = \{S \in \mathbb{R}^{m\times n}, \|S\|_0 \leq k\}$, the projection $P_{\mathcal{E}_j}$ keeps k coefficients with the largest absolute values and sets other coefficients to zero. Since the gradient step in the update rule of PALM (2) cannot dramatically change the coefficients, the k indices with the largest absolute value are very likely the same as in the previous iteration, in which case the support of the solution *does not evolve* during the iterations of PALM. Empirical experiments confirm that the support can indeed remain unchanged forever right after the first iteration.

To overcome this lack of support exploration, we propose to adapt Hard Thresholding Pursuit [9] (HTP), an algorithm addressing the sparse linear regression problem [14]:

$$\begin{aligned} \underset{x\in\mathbb{C}^n}{\text{minimize}} \quad & \|Bx - y\|^2 \\ \text{subject to:} \quad & \|x\|_0 \leq s. \end{aligned} \tag{3}$$

The problem involves the ℓ_0 constraint, which requires an algorithm to search for the set of nonzero coefficients, or the support of the optimal solution. This similarity between our problem and problem (3) sparks the idea to adapt HTP to matrix factorization. Our attempt is to firstly deal with the simple case of 2-factor factorization, formulated as:

$$\begin{aligned} \underset{X\in\mathbb{R}^{m\times n}, Y\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad & \|A - XY\|^2 \\ \text{subject to:} \quad & X \in \mathcal{E}_X, Y \in \mathcal{E}_Y \end{aligned} \tag{4}$$

where $\mathcal{E}_X, \mathcal{E}_Y$ are some structured sparse matrix sets. Since both X and Y need to be optimized, to make the setting similar to problem (3), we fix one matrix, and optimize the other. When Y is fixed, the rule to update X is:

$$\begin{aligned} R_{k+1} &\leftarrow \texttt{supp}(P_{\mathcal{E}_X}(X^k + \mu_X(A - X_kY_k)Y_k^T)) \\ X_{k+1} &\leftarrow \arg\min_X\{\|A - XY_k\|_2, \texttt{supp}(X) \subseteq R_{k+1}\} \end{aligned} \tag{5}$$

with supp the support (set of nonzero coefficients) of a matrix.

Algorithm 1 Bilinear Hard Thresholding Pursuit

```
procedure BHTP(A, E_X, E_Y, μ_X, μ_Y)
    while terminating condition is not met do
        Fix Y (resp X) and optimize X (resp Y) with (5).
        for i = 1, ... k do
            Use equation (2) to optimize X, Y.
        end for
    end while
    Return X, Y.
end procedure
```

Alternatively fixing and optimizing X and Y with equation (5) is supposed to force the support exploration explicitly. The update rule (2) can be employed between iterations to polish the solution of the current support. The resulting algorithms, called Bilinear Hard Thresholding Pursuit (BHTP), is summarized in Algorithm 1.

3. GENERALIZED HUNGARIAN METHOD

The Hadamard Transform H_n of size 2^n (and matrices with similar structure [15]) can be defined recursively as:

$$H_0 = 1, H_{n+1} = \frac{1}{\sqrt{2}}\begin{pmatrix} I_n & I_n \\ I_n & -I_n \end{pmatrix}\begin{pmatrix} H_n & 0_n \\ 0_n & H_n \end{pmatrix}. \tag{6}$$

Further expansion of this recursive formula reveals that H_n is a product of n factors, each with exactly 2-nonzero coefficients *per row and per column*. Therefore, in [11], $\mathcal{E}_k^c = \{\|S_{i,\bullet}\|_0 \leq k, \forall i\}$ and $\mathcal{E}_k^c = \{\|S_{\bullet,i}\|_0 \leq k, \forall i\}$ are considered ($S_{i,\bullet}, S_{\bullet,i}$ are the i^{th} row and column of S respectively). However, the proximal operators corresponding to $\mathcal{E}_k^r$ (resp $\mathcal{E}_k^c$) can produce matrices with vanishing columns (resp rows), which are not the true factors of H_n.

In [11] and the corresponding code[1] this problem is addressed by the usage of an operator named SPLINCOL : given an input matrix, it outputs a matrix where all coefficients are set to zero except on a support defined as the union of the supports produced by the projections on $\mathcal{E}_k^r$ and $\mathcal{E}_k^c$ for this input matrix. SPLINCOL is however not a proximal operator, and it can produce rank-deficient matrices, which is also undesirable. Hence, we would like to define a set $\mathcal{E}$ and a proximal operator that well describe the sparse structure of the Hadamard Transform and avoid the problems of $\mathcal{E}_k^r$ and $\mathcal{E}_k^c$. This gives birth to the definition of **k-regular sparse matrices**.

Definition 3.1. *A k-regular sparse matrix $U \in \mathbb{R}^{n\times n}$ (or $\mathbb{C}^{n\times n}$) is a matrix whose columns and rows contain at most k nonzero entries each. The set of such matrices is denoted $\mathcal{R}_k$.*

3.1. Pitfall with the HF approach

Consider factorizing the Hadamard tranform of dimension 8, $H_3 = S_1S_2S_3$, $S_i \in \mathcal{R}_2$. In an ideal scenario, HF would firstly factorize $H_3 = \hat{S}_1S_2^*$, then $S_2^* = \hat{S}_2\hat{S}_3$ and finally get $\hat{S}_i = S_i$ (possibly up to natural permutation and scaling ambiguities). Given the simple observation that the product of two matrices in $\mathcal{R}_2$ is always in $\mathcal{R}_4$, a natural approach to favor the idealized behavior of HF would be to perform its first step using PALM on H_3 with $\mathcal{E}_1 = \mathcal{R}_2$ and $\mathcal{E}_2 = \mathcal{R}_4$. However it is possible to show that there exists $\hat{S}_1 \in \mathcal{R}_2, S_2^* \in \mathcal{R}_4$ such that $H_3 = \hat{S}_1S_2^*$ but S_2^* cannot be further factorized into $\hat{S}_2, \hat{S}_3 \in \mathcal{R}_2$. This indicates that in certain settings, directly addressing the multilayer factorization problem with PALM (rather than with HF) is desirable if possible.

3.2. An algorithm to project on $\mathcal{R}_k$

To project on the set $\mathcal{R}_k$ one can exploit the following lemma:

Lemma 3.1. *Let I be the collection of all sets $\mathcal{I} \subset \{1, \ldots, n\}^2$ of $n \times k$ matrix indices with each row and column containing exactly k elements. Given an $n \times n$ matrix U, let $\mathcal{I} \in I$*

[1] https://faust.inria.fr

be a set maximizing $\sum_{(i,j)\in\mathcal{I}} |U_{ij}|^2$ among such sets. The projection of U onto $\mathcal{R}_k$ is:

$$P_{\mathcal{E}}(U) = U_{\mathcal{I}} \tag{7}$$

where $U_{\mathcal{I}}$ is the matrix whose entries match those of U on $\mathcal{I}$ and are set to zero elsewhere.

Denoting $c_{ij} = -|U_{ij}|^2$, finding $\mathcal{I}$ is reducible to the following integer problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{1\le i,j\le n} c_{ij}x_{ij} \\ \text{subject to:} \quad & \sum_{j=1}^{n} x_{ij} = k \qquad \forall i = 1,n \\ & \sum_{i=1}^{n} x_{ij} = k \qquad \forall j = 1,n \\ & x_{ij} \in \{0,1\} \qquad \forall 1 \le i,j \le n \end{aligned} \tag{8}$$

When $k = 1$, problem (8) becomes the classic assignment problem, which is efficiently solved with the Hungarian method [16]. To adapt the idea of the Hungarian method for $k > 1$ let us first proceed with an important definition:

Definition 3.2 (k-disjoint perfect matching and its value)**.** *Let $G = (V, E)$ be a complete weighted bipartite graph where $V = S \cup T, |S| = |T|$ and weight function $c : S \times T \to \mathbb{R}$. A k-disjoint perfect matching $M \subset E$ is a disjoint union of k perfect matchings. The value of a k-disjoint perfect matching M is $V(M) := \sum_{e\in M} c(e)$.*

Problem (8) corresponds to **minimizing** the value of a k-disjoint perfect matching of a bipartite graph whose weight are c_{ij}. The Hungarian method is a primal dual algorithm [17], which reduces the dual gap between dual and primal problem. The dual problem is formulated as follows:

Definition 3.3 (Potential and its value)**.** *Let $G = (V, E)$ be a complete weighted bipartite graph where $V = S \cup T, |S| = |T|$ and weight function $c : S \times T \to \mathbb{R}$. A potential π is a function $f : V \to \mathbb{R}$. A potential π is feasible if $\forall v \in V, |\{u \in V | c_{vu} < \pi(u) + \pi(v)\}| \le k$. The value of π is $P(\pi) := k(\sum_{v\in V} \pi(v)) + \sum_{(u,v)\in E} \min(0, c_{uv} - \pi(u) - \pi(v))$.*

The dual of the problem of **k-disjoint perfect matching value minimization** is **maximization of the value of a potential**. To describe the resulting algorithm, we introduce a few additional definitions.

Definition 3.4 (Graph built from a potential)**.** *Let $G = (V, E)$ be a complete weighted bipartite graph where $V = S \cup T, |S| = |T|$ and weight function $c : S \times T \to \mathbb{R}$ and π be a feasible potential. Define:*

1. $E_< = \{u, v \in V | c(u, v) < \pi(u) + \pi(v)\}$
2. $E_= = \{u, v \in V | c(u, v) = \pi(u) + \pi(v)\}$

Let $E_\pi = E_< \cup E_=$. The graph $G_\pi = (V, E_\pi)$ built from π is a subgraph of G having edge set E_π.

Definition 3.5 (k-saturated vertex)**.** *Let $G = (V, E)$ be a complete weighted bipartite graph where $V = S \cup T, |S| = |T|$, weight function $c : S \times T \to \mathbb{R}$ and a set of edges $M \subset E$. A vertex v in V is k-saturated w.r.t M (or k-saturated in short) if there are exactly k edges in M having v as their endpoints. Otherwise, the vertex is k-unsaturated.*

Definition 3.6 (Alternating and augmenting path)**.** *Let $G = (V, E)$ be a bipartite graph where $V = S \cup T, |S| = |T|$. Let $M \subset E$ be a subset of graph edges. An alternating path $(e_1, \ldots, e_L)$ satisfies that $e_i \in M \iff i \mod 2 = 0$. An augmenting path $(e_1, \ldots, e_{2n+1})$ of M is an alternating path starting and ending with k-unsaturated vertices.*

An augmentation of M with respect to an augmenting path $(e_1, \ldots, e_{2n+1})$ is $M \backslash \{e_2, e_4, \ldots, e_{2n}\} \cup \{e_1, e_3, \ldots, e_{2n+1}\}$.

Algorithm 2 Generalized Hungarian Method

procedure GHM($G = (S \cup T, E), c, k$)
  Initialize $\pi(u) = \min_{i,j} c_{ij}, u \in S, \pi(v) = 0, v \in T$
  Build bipartite graph $G_\pi = (S \cup T, E_< \cup E_=)$ w.r.t π
  Initialize $M = E_<$.
  while M is not a k-disjoint perfect matching **do**
    Let Z be the set of vertices reachable from a k-unsaturated vertex in S by an alternating path.
    if Z has a k-unsaturated vertex $t \in T$ **then**
      Perform augmentation of M.
      continue
    end if
    Let $S_1 = S \cap Z, S_2 = S \setminus Z$.
    Let $T_1 = T \cap Z, T_2 = T \setminus Z$.
    Let $r(u, v) = c(u, v) - \pi(u) - \pi(v), (u, v) \in E$.
    Let $\sigma_1 = \min_{(u,v)\in S_1\times T_2} \{r(u, v) \mid r(u, v) > 0\}$.
    Let $\sigma_2 = \min_{(u,v)\in S_2\times T_1} \{-r(u, v) \mid r(u, v) < 0\}$.
    Let $\Delta = \min(\sigma_1, \sigma_2)$. Increase $\pi(u)$ by Δ for $u \in S_1$, decrease $\pi(v)$ by Δ for $v \in T_1$.
    Rebuild G_π from the new potential.
  end while
  Return M**.**
end procedure

Algorithm 2, referred to as Generalized Hungarian Method (GHM), is essentially a specification of the primal dual min cost flow algorithm [18] for a specific bipartite graph. Its overall complexity is $\mathcal{O}(kn^4)$. When $k = 1$, this becomes $\mathcal{O}(n^4)$, the raw complexity of the Hungarian method. There exists a dynamic programming version of the Hungarian method to improve the worst case complexity to $\mathcal{O}(n^3)$. Adapting this idea leads to Algorithm 3 reducing the complexity to $\mathcal{O}(kn^3)$.

Algorithm 3 An optimized augmentation

procedure OPTIMIZED_AUGMENTATION(G, c)
  Initialize $inZ[v] =$ **FALSE**, $\forall v \in V$.
  $inZ[u] =$ **TRUE** for $u \in S$ k-unsaturated.
  Initialize a queue Q, $u \in Q$ if $u \in S$ k-unsaturated.
  Initialize $\sigma(u) =$ **INF**, $u \in V$.
  Initialize $\Delta = 0$
  while No augmenting path in $G' = (V, E_=)$ **do**
    Perform Breath First Search [19] using vertices in Q for graph G' with inZ memorizing discovered vertices.
    Update $\sigma(u) = \min_{r(u,v)>0}(\sigma(u) - \Delta, r(u,v)), u \in T_2$ with newly added $v \in S_1$.
    Update $\sigma(u) = \min_{r(u,v)<0}(\sigma(u) - \Delta, -r(u,v)), u \in S_2$ with newly added $v \in T_1$.
    Calculate $\Delta = \min_{u \in T_2 \cup S_2} \sigma(u)$.
    Update potential for $u \in S_1 \cup T_1$ (Algorithm 2).
    Add the set $U = \{u | \sigma(u) = \Delta\}$ to Q.
  end while
  Perform augmentation similar to Algorithm 2.
end procedure

4. EMPIRICAL EXPERIMENTS

Two experiments illustrate the effectiveness of our proposals.

4.1. BHTP and factorization of magnetoencephalography (MEG) matrix

We measure the performance of BHTP and PALM in the context of functional brain imaging with magnetoencephaloraphy (MEG) signal. It is the product between an original functional signal of high dimension and the MEG gain matrix $A \in \mathbb{R}^{8193 \times 204}$ (obtained by NME software [20]). Sparse factorization of A can significantly reduce the computation cost to solve the inverse functional signal problem [21]. We verify the impact of BHTP by comparing its performance with PALM to address the approximation problem (4) with A the MEG matrix, $\mathcal{E}_X = \mathcal{E}^r_{k_0} \cap \mathbb{R}^{8193 \times 204}$, $\mathcal{E}_Y = \mathcal{E}^c_k \cap \mathbb{R}^{204 \times 204}$ for $k_0 \in \{25, 75, 50, 100\}, k \in \{25, 50, 75, 100, 125, 150, 175, 200\}$, $\mu_X = 10^{-3}, \mu_Y = 10^{-4}$. Denoting $\hat{X}, \hat{Y}$ the estimated factors, we used the following performance measure:

$$ER = \frac{\|A - \hat{X}\hat{Y}\|_F}{\|A\|_F} \tag{9}$$

The results are illustrated in Figure 1. We observe that BHTP consistently achieves better error than PALM when $k_0 \in \{75, 100\}$. In other cases, BHTP is better when $k_1 \leq 75$ and the error gap is huge especially with small k_1 (i.e, $k_1 = 25$). It can be explained as the sparser is the matrix, the more important is the search for support (since their coefficients contribute more). Therefore, BHTP with mechanism to avoid premature convergence of its supports will be less affected than PALM.

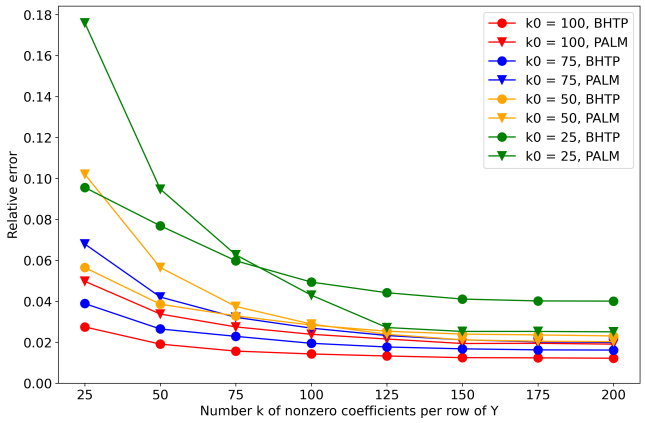

Fig. 1: Performance of BHTP and PALM.

4.2. The Hadamard Transform Factorization and GHM

To factorize the Hadamard Transform one can either use PALM and HF as in [11], or exploit the 2-regular property of its factors using PALM with the same protocol (initialization, learning rate, etc.) but with SPLINCOL replaced by GHM. As documented in [11], PALM implemented with SPLINCOL fails to factorize the Hadamard transform ($ER = 0.85$), while HF succeeds thanks to taylored structured sparsity patterns $\mathcal{E}_X, \mathcal{E}_Y$ (using SPLINCOL) for intermediate two-factor problems. In contrast, exploiting the proposed GHM projection operator within PALM yields an exact factorization ($ER = 2 \times 10^{-16}$). Adapting HF to use GHM on the left factor for intermediate two-factor problems also yields exact factorization ($ER = 2 \times 10^{-16}$).

Even though we showed there exists non-factorizable intermediate matrices, which would prevent HF from achieving global factorization, the default initialization of [11] apparently avoids such a scenario. Nevertheless, the ability of PALM with GHM to directly address the global factorization problem without this potential weakness of HF seems advantageous and has the potential of making it more robust to initialization.

5. CONCLUSION

Besides introducing the set of k-regular matrices and describing a GHM algorithm to project a matrix onto this set, we defined BHTP, an algorithm to address the premature convergence of matrix supports in bilinear sparse matrix factorization. Numerical experiments illustrate the effectiveness of our proposals. Beyond extending BHTP to the multilinear/multifactor and/or complex-valued case, some challenges lying ahead involve better understanding how to setup stepsizes for the algorithms, how to harness intrinsic scaling ambiguities of the problems, and how to speedup the algorithms to handle very large matrices.

6. REFERENCES

[1] E. M. L. Beale, M. G. Kendall, and D. W. Mann, "The discarding of variables in multivariate analysis," *Biometrika*, vol. 54, no. 3/4, pp. 357–366, 1967.

[2] R. R. Hocking and R. N. Leslie, "Selection of the best subset in regression analysis," *Technometrics*, vol. 9, no. 4, pp. 531–540, 1967.

[3] Arthur E. Hoerl and Robert W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.

[4] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[5] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[6] S. CHEN, S. A. BILLINGS, and W. LUO, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44 vol.1.

[8] Stéphane Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[9] Simon Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM J. Numerical Analysis*, vol. 49, pp. 2543–2563, 01 2011.

[10] James Cooley and John Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[11] L Le Magoarou, Rémi Gribonval, and 2016, "Flexible multilayer sparse approximations of matrices and applications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 688–700, 2016.

[12] Neal Parikh and Stephen Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[13] Jerome Bolte, Shoham Sabach, and Marc Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[14] Trevor Hastie, Robert Tibshirani, and Martin Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman & Hall/CRC, 2015.

[15] Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Ré, "Learning fast algorithms for linear transforms using butterfly factorizations," *CoRR*, vol. abs/1903.05895, 2019.

[16] H. W. Kuhn and Bryn Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart*, pp. 83–97, 1955.

[17] Stephen J. Wright, *Primal-Dual Interior-Point Methods*, Society for Industrial and Applied Mathematics, USA, 1997.

[18] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver, *Combinatorial Optimization*, John Wiley & Sons, Inc., USA, 1998.

[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms, Third Edition*, The MIT Press, 3rd edition, 2009.

[20] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti Hämäläinen, "Mne software for processing meg and eeg data," *NeuroImage*, vol. 86, 10 2013.

[21] Luc Le Magoarou, Rémi Gribonval, and Alexandre Gramfort, "FAμST: Speeding up linear transforms for tractable inverse problems," in *EUSIPCO - 23rd European Signal Processing Conference*, 2015.